

Optimización de Arquitectura de Google Course Builder para Google AppEngine

Cada día hay más personas que usan plataformas digitales para aumentar sus conocimientos, entre ellos, las plataformas MOOC. Estas plataformas reciben grandes cantidades de estudiantes concurrentes y por lo tanto, necesitan cierta escalabilidad y eficiencia para poder manejar tanto tráfico.

Existen ciertas plataformas tecnológicas como Moodle, Sakai, Edx y Course Builder que ofrecen las funcionalidades tecnológicas necesarias para una plataforma MOOC, pero no suelen venir preparados para la gran cantidad de usuarios y cálculo de estadísticas que caracterizan a estas plataformas. Un gran problema para estas plataformas es su rentabilidad, y los altos costes de servidor, provocados por la mala arquitectura y diseño de las mismas.

En este trabajo voy a tratar de explicar los problemas de arquitectura que tiene Google Course Builder, problemas de modularidad, diseño y manejo de datos, y voy a proponer cambios para una nueva arquitectura que soluciona los problemas enumerados.

El caso de estudio para el proyecto es UniMOOC, una plataforma de cursos online de la Universidad de Alicante que usa la tecnología de Google Course Builder, y del cual soy el director de tecnología. UniMOOC sufrió un gran aumento de registros de usuarios y con ello, un aumento drástico en los costes de servidor. Si no cambiaba algo rápido, la plataforma iba a morir de éxito.

Tras estudiar la funcionalidad de Google AppEngine, la plataforma de hosting de aplicaciones del Cloud Platform de Google, vi como el uso de cache, una actualización del tipo de modelos de la base de datos, y ciertos cambios en la configuración de la aplicación podían mejorar considerablemente la eficiencia de la plataforma y por lo tanto, reducir los costes de servidor. El uso de cache en las transacciones de la base de datos aumenta el rendimiento del acceso de datos, reduciendo la cantidad de instancias que lanza AppEngine para hacer frente a las peticiones que llegan de los usuarios. Esta reducción de instancias es el mayor ahorro en costes de servidor.

Además de los cambios de rendimiento y eficiencia, se ha mejorado la modularidad del código del sistema software para facilitar el desarrollo de nuevas funcionalidades y mejorar el mantenimiento del sistema.

Los resultados de estos cambios se han estudiado durante un periodo de 10 meses sobre UniMOOC y demuestran la eficacia de los cambios realizados, habiendo rebajado los costes de servidor un 77% y los accesos a la base de datos un 94%. Estas mejoras han permitido que UniMOOC siga creciendo, llegando en estos momentos a tener más de 350.000 estudiantes y unos 30 cursos distintos de formación.